

REMARKS

In an Office Action dated June 21, 2006, the Examiner rejected claim 13 under 35 U.S.C. §101 as directed to non-statutory subject matter; rejected claim 1-18 under 35 U.S.C. §112, second paragraph, as indefinite, rejected claims 1, 2, 4, 6-9, 11-14, 16 and 18 under 35 U.S.C. §102(b) as anticipated by Cederqvist et al., “Version Management with CVS” (herein *Cederqvist*); rejected claims 3 and 15 under 35 U.S.C. 103(a) as unpatentable over *Cederqvist* in view of “Introduction to the vi Editor”, Rice University 2002 (herein *Rice*); and rejected claims 5, 10 and 17 under 35 U.S.C. §103(a) as unpatentable over *Cederqvist*

Amendment to Specification

Applicants have made a minor amendment to the Specification to correct an erroneous reference number. No new matter is introduced.

Non-Statutory Subject Matter

The Examiner does not state clearly the basis for the non-statutory subject matter rejection, but applicants understand the Examiner’s objection to be grounded on the possible interpretation of “signal-bearing media” to include intangible transmission media. Applicants have accordingly amended claim 13 to recite “tangible signal-bearing media”. Computer instructions recorded on tangible media have been recognized as a apparatus and therefore statutory subject matter.

Indefiniteness

All surviving claims have been amended to change “encoded instructions” and “encoded statements” to refer specifically to source code of a computer program. This appears to be consistent with the Examiner’s interpretation for purposes of examination and is sufficiently definite. Because the independent claims now specifically recite computer programming source code, dependent claims 2, 9 and 14 have been cancelled as superfluous.

Prior Art

Applicants have amended all independent claims to clarify the claimed invention. In particular, the claims have been amended to clarify that the statements being copied are source code statements within a source code file, and that all the copies exist together at different locations in the file and are compiled together into the object code. As amended, the claims are patentable over the cited art.

A brief background discussion is in order to appreciate applicants' invention. Applicant's invention lies in the realm of user interface, and is intended to improve an interactive interface between a computer programmer and a source code editor. Frequently, a programmer copies individual statements or small snippets of code from one location in the source file to another. It may subsequently be discovered, e.g. during a compilation process, that some syntax or other error exists in the original statement from which the copies were made. If multiple copies were made, the programmer must manually edit each such copy.

In accordance with applicants' invention, a record of such copies is kept, and a programmer may correct all such copies at once by indicating that a manually edited change is to be automatically propagated to the various copies. Thus, the need to manually edit each copy of the statement is removed.

Cederqvist discloses a versioning control system for developing computer software. As is well known in the art, computer software is often developed by teams of individuals who work concurrently on multiple modules of source code. In these circumstances, there is a need to maintain coherency. I.e., with respect to any particular source code module, different developers may simultaneously wish to edit different portions of the module. *Cederqvist* discloses a system whereby changes made by different developers are tracked and can be introduced into a master copy ("committed") independently.

The Examiner's rejection, as nearly as applicants can understand it, appears to read applicants' recited statements on the source files themselves. I.e, if multiple versions ("copies") of a source file exist, then changes to one can be propagated automatically to another using *Cederqvist*'s versioning mechanism.

While applicant's do not necessarily agree with this reading of their claims, in the interests of furthering prosecution herein they have amended all independent claims to clarify that the claims relate to multiple copies of statements *within the same source file*. Furthermore, these copies are not replacements for one another (different versions of one another), but are *independent source code statements at different locations in the source code, which are compiled together into the object code forming the computer program*.

Applicants' representative amended claim 1 recites:

1. A method for developing source code for a computer program, comprising the steps of:
 - generating a plurality of source code statements in a source code file, said source code file being compilable into object code of said computer program;
 - automatically maintaining a record of status of each respective source code statement;
 - editing a first source code statement of said plurality of source code statements to produce an edited first source code statement;
 - automatically determining whether one or more copies of said first source code statement exist within said source code file from said status of each respective source code statement, *each of said first source code statement and copy of said first source code statement occupying a different respective location within said source code file and being compilable together into said object code of said computer program*; and
 - responsive to said automatically determining step, automatically propagating changes made by said editing step to said one or more copies of said first source code statement.

[emphasis added]

Claim 13 is a program product claim reciting analogous steps. Claim 7 is a method claim directed to the aspect of performing a verification action (such as requesting confirmation from a user) before copying a source code statement which has not been previously verified. All of these claims deal with copying of source code statements within a source code file, so that multiple

independent statements at different code locations are created (although they may have originally been copied from a common statement created by the user).

Cederqvist relates to copies of *files*. It is true that these files contain source code statements, but the files themselves are different versions of the same code. I.e., once copied, they are not independently functioning entities at different locations within a source code file, which can be compiled together. Only one of the multiple versions of the source code file is meant to be compiled at any one time, and only one can be included in the program at any one time. The existence of multiple versions is a result of the complicated history of development, the fact that multiple people are involved, and similar factors. It is not because each version of the same source code module is compiled together and included in the computer program. Such a thing would make no sense, and lead to no end of errors in the final product.

Cederqvist therefore does not disclose the copying of a source code statement *from one location in a source code file to another location in the same source code file*, tracking the multiple *statements at different locations in the same source file* which were originally derived from a single statement, where the *multiple statements are compiled together into the object code*, and automatically propagating a change made in one of the statements to the other statements.

For all of these reasons, the claims as amended are not anticipated by *Cederqvist*. Nor are the claims obvious over *Cederqvist*, either alone or in combination with *Rice*. As explained above, *Cederqvist* relates to an entirely different problem of managing multiple different versions of source modules being edited by different programmers. There is nothing that would suggest tracking a user copying a statement of code from one location in his source to another, so that subsequent changes to the original statement could be propagated to the statement derived from it. Nor is there anything that would suggest taking a verification action before a user copies an

unverified statement. Both of these features have nothing to do with controlling many versions of a source module which can be spawned by a complex development history involving different developers, as in the case of *Cederqvist*'s system. They are, rather, intended to improve the interactive efficiency of the programmer with the source editor (and as such are equally applicable whether code is being edited by a single individual or multiple individuals). *Rice* is cited to show the feature of requesting user confirmation of a change in a text editor as recited in certain dependent claims, but does not otherwise teach or suggest the crucial features of applicants' invention.

In view of the foregoing, applicants submit that the claims are now in condition for allowance, and respectfully request reconsideration and allowance of all claims. In addition, the Examiner is encouraged to contact applicants' attorney by telephone if there are outstanding issues left to be resolved to place this case in condition for allowance.

Respectfully submitted,

CARY L. BATES, et al.



By: _____

Roy W. Truelson
Registration No. 34,265

Telephone: (507) 202-8725

Docket No.: ROC920030211US1
Serial No.: 10/720,961